læringsmålene for denne workshop

1. De studerende skal kunne lave et git repository og clone et fra et remote repository.
2. De studerende skal kunne tilføje filer til deres repository, og committe ændringer i disse filer.
3. De studerende skal kunne dele deres lokale ændringer med et online repository
4. De studerende skal kunne håndtere konflikter ved merge.
5. De studerende skal have forståelse for hvad GIT og GITHUB er.
6. De studerende skal kunne administrere deres online repo gennem GITHUB.

Git kommandoer de studerende skal kunne anvende efterfølgende:

1. Init og Clone
2. Add og commit
3. Push og Pull
4. Branch/checkout og merge

!! Er der problemer med ’unrelated histories’ så kan følgende bruges til at hente

Git pull --allow-unrelated-histories

# Hvorfor bruge versionsstyring?

Og hvorfor lige GIT?

Med GIT for de studerende mulighed for at dele deres kode med deres gruppemedlemmer, nemt og sikkert.

**Nemhed**, kommer af når først projektets repository er opsat, så er der meget få kommandoer der skal bruges for at kunne dele sine ændringer med alle andre i gruppe.

**Sikkerhed**, kommer af at GIT holder styr på alle ændringer der er blevet lavet, så hvis man husker at gøre brug af sine commits, så kan man altid gå tilbage hente gamle tilstande tilbage.

**Gruppearbejde**: GIT er yderst vigtigt for de studerende, eftersom de kommer til at skulle lave mange projekter, hvori de forskellige gruppemedlemmer skal kunne arbejde så selvstændigt som muligt og samtidigt have muligheden for at få de seneste ændringer flettet ind på den nemmeste måde.

Derudover så bliver det også bare et krav til senere eksamener at i kan aflevere et projekt som et repository via GITHUB. Og det er en standard for 99.99% af virksomheder (der udvikler software).

# Opstart af et projekt

Der er flere forskellige måder at start ud på, alt efter om der allerede findes et repository der skal gøres brug af, eller der allerede findes kode som skal deles ud på et nyt repository.

## GIT INIT

# Kom godt i gang

For at kunne bruge git og github er der en række ting der skal på plads først.

1. Have installeret GIT -> <https://git-scm.com>
2. Have en konto på GITHUB -> [www.github.com](http://www.github.com)

# Opgaver 1 – Git clone

Opgaven går ud på at du får forståelse for hvordan du henter et allerede eksisterende repository, ned på din computer, hvorfra du selv kan foretage ændringer.

1. Lav et Repository på Github.
2. Ved brug af Git Bash klon dit repository, så du har det lokalt på din computer.
3. Foretag ændringer i den mappe Git har oprettet for dig.
   1. F.eks. kan du lave en java file, med en main metode der udskriver Hello World til konsolen. Hvad filen indeholder er ikke vigtigt, bare du har en fil du kan ændre i.
4. Commit dine lokale ændringer.
5. Skub dine ændringer op på github.

# Opgave 2 – Git + Grupper

Opgaven går ud på at i, som gruppe, kan dele kode mellem jer, igen ved brug af et fælles repository på github.com

1. En i gruppen laver et repository (eller bruger det fra opgave 1)
2. Den der lavede et repository skal tilføje alle gruppemedlemmer som ’Collaborators’
   1. Under ’settings’ er der en tab der hedder ’Collaborators’, her skal alle gruppes medlemmers mail angives.
3. Når alle er tilmeldt som collaborators, så skal alle klone dette ene repository til hver sin computer.
4. Hvert medlem skal oprette en ny fil i sit lokale repository og dele den med de andre.
   1. Til at starte med kan det være en meget god ide at hver fil har forskellige navne.

# Opgave 3 – Git + Conflicts

Opgave går ud på at du skal kunne håndtere ’merge conflicts’.

1. Fortsæt med det repository i gjorde brug af i opgave 2.
2. Nu hvor alle i gruppe har alle filer, så skal i begynde at rette i den sammen fil.
   1. Ændringer skal også commits til det lokale repository.
3. En i gruppe skal så skubbe sine ændringer op på github.
4. Alle andre i gruppe skal derefter hente disse ændringer og løse de konflikter der kan forekomme.
5. Til sidst skal man gerne (på github) se alle gruppemedlemmers ændringer i den samme fil.

# Opgave 4 – Git Init – EKSTRA OPGAVE (!!)

Opgave går ud på at bruge en lokal kodebase som startpunkt, og derved skubbe al data i den ønskede mappe op på github uden at skulle klone et repository.

1. Lav et repository på github.
2. Lav en mappe med navnet tProjekt og lav en fil i mappen som hedder HelloWorld.java
3. Fyld filen ud med en klasse, af samme navn, og lav samtidig en main metode
4. Initialiser mappen til brug med git.
   1. Git init, initialiserer den mappe du står i.
5. Skub nu filen på dit github repository.
   1. Når du har lave et commit og kalder git push, så skal du blot følge vejledningen git selv giver.